

WHAT IS CLAIMED IS:

1. A method of determining whether an electronic device is simulated, comprising:
 - storing a first value in a memory location;
 - executing at least one operation on said
 - electronic device wherein said at least one
 - operation causes said memory location to change
 - value if said electronic device is simulated and
 - wherein said at least one operation does not cause
 - said memory location to change value if said
 - electronic device is not simulated; and
 - reading a second value from said memory
 - location to determine whether said second value is
 - different from said first value.
2. The method of claim 1, wherein said electronic device comprises a computer.
3. The method of claim 1, wherein said memory location comprises a register.
4. The method of claim 3, wherein said register comprises general register 0.
5. The method of claim 3, wherein said register comprises a return register.
6. The method of claim 1, wherein said at least one operation comprises a ldil operation.
7. The method of claim 1, wherein said at least one operation comprises at least one nop operation.

8. The method of claim 7, wherein each of said at least one nop operation is followed by a parameter.
9. A method of indicating that an electronic device is a simulation, comprising:
 - comparing instructions for said electronic device with a predetermined sequence of instructions; and
 - indicating that said electronic device is a simulation if said instructions include said predetermined sequence of instructions.
10. The method of claim 9, wherein said electronic device comprises a computer.
11. The method of claim 9, wherein indicating that said electronic device is a simulation comprises storing a value in a memory location.
12. The method of claim 9, wherein said indicating that said electronic device is a simulation is performed only if said predetermined sequence of instructions in said instructions are not interrupted by another instruction.
13. The method of claim 9, wherein said predetermined sequence of instructions comprises at least one operator and parameter.
14. The method of claim 13, wherein said at least one operator and parameter comprises at least one ldil operator and a predetermined number.

15. The method of claim 13, wherein said at least one operator and parameter comprises at least one nop operator and a predetermined number.

16. The method of claim 9, wherein said predetermined sequence of instructions comprises a sequence of null operations.

17. A computer hardware simulator, comprising:

a) at least one computer readable storage medium; and

b) computer readable program code stored in the at least one computer readable storage medium, the computer readable program code comprising:

i) code for receiving instructions for said computer hardware;

ii) code for comparing said instructions with a predetermined set of instructions; and

ii) code for setting a flag if said instructions contain said predetermined set of instructions, wherein said flag indicates that said computer hardware is simulated rather than physical.

18. The computer hardware simulator of claim 17, wherein said flag comprises a predetermined value stored in a register.

19. The computer hardware simulator of claim 17, wherein said predetermined set of instructions must be included in said instructions in order with no other intervening instructions before setting said flag.

20. The computer hardware simulator of claim 17, wherein said predetermined set of instructions comprises a plurality of ldil operations storing predetermined values to general register 0.
21. The computer hardware simulator of claim 17, wherein said predetermined set of instructions comprises a plurality of nop operations with predetermined parameters.
22. A computer hardware simulator, comprising:
means for receiving an instruction sequence;
means for comparing said instruction sequence with a predetermined instruction sequence; and
means for indicating that said computer hardware is a simulation if said instruction sequence matches said predetermined instruction sequence.
23. A method of programming a computer, comprising:
writing a computer program for said computer;
compiling said computer program to create an executable program;
executing said executable program on a simulation of said computer; and
executing said executable program on said computer.

5

compiling said computer program to create an executable program which may be executed both on said computer and on a simulation of said computer.

[illegible]